Staleness vs. Waiting Time in Universal Discrete Broadcast

Michael Langberg Alexander Sprintson Jehoshua Bruck
California Institute of Technology
Email: {mikel,spalex,bruck}@caltech.edu

***Abstract*— In this paper we study the distribution of dynamic data over a broadcast channel to a large number of passive clients. The data is simultaneously distributed to clients in the form of *discrete* packets, each packet captures the most recent state of the information source. Clients obtain the information by accessing the channel and listening for the next available packet. This scenario, referred to as *discrete broadcast*, has many practical applications such as the distribution of stock information to wireless mobile devices and downloading up-to-date battle information in military networks.**

Our goal is minimize the amount of time a client has to wait in order to obtain a new data packet, i.e., the *waiting time* of the client. We show that we can significantly reduce the waiting time by adding redundancy to the schedule. We identify *universal* schedules that guarantee low waiting time for any client, regardless of the access pattern.

A key point in the design of data distribution systems is to ensure that the transmitted information is always up-to-date. Accordingly, we introduce the notion of *staleness* that captures the amount of time that passes from the moment the information is generated, until it is delivered to the client. We investigate the fundamental trade-off between the staleness and the waiting time. In particular, we present schedules that yield lowest possible waiting time for any given staleness constraint.

I. Introduction

Modern society has become heavily dependent on wireless networks in order to deliver information to diverse clients. People expect to be able to access the latest data, such as stock quotes and traffic conditions, at any time, whether they are at home, in an office, or traveling. Wireless data distribution systems also have a broad range of applications in military networks, such as transmitting up-to-date battle information to tactical commanders in the field. New applications place high demands on the quality, availability, and timeliness of data distribution.

An important characteristic of wireless infrastructure is the asymmetry between the downlink and uplink channels. The downlink channel is of much higher bandwidth than the uplink channel. Moreover, while the downlink channel is operated by a powerful antenna, the uplink channel is driven by a mobile device with limited power resources.

This intrinsic asymmetry of wireless infrastructure impacts the way information is delivered to clients. In particular, the standard *client-server* paradigm, in which the data transfer is initiated by clients, may not be adequate for some wireless systems. Wireless data broadcast [1]–[3] has recently emerged as an attractive way to disseminate data to a large number of clients. In data broadcast systems, the server proactively transmits the information on the downlink channel and the clients access data by listening to the channel. Wireless data broadcast is an attractive way to deliver dynamic data such as stock quotes, popular web pages, and traffic conditions. This approach enables the system to serve a large number of heterogenous clients, minimizing power consumption and keeping the clients' locations secret.

Fig. 1 depicts a typical data broadcast system. The system includes the following components: the database, the server (scheduler), the broadcast channel, and the wireless clients. The server periodically accesses the database, retrieves the most recent data, encapsulates it into packets and sends the packets (or encoding thereof) over the broadcast channel.

A key challenge in the design of systems for wireless data broadcast is to identify an *optimum schedule*, i.e., a time sequence that specifies the "best" data to transmit over the channel at any point in time. The schedule must minimize both the *waiting time* of the client and the *staleness* of the information. Waiting time is the amount of time spent by a client waiting for data. The waiting time is an extremely important parameter for many applications. In addition, it is closely related to the amount of power spent by the client to obtain the information. The staleness captures the amount of time that passes from the moment the information is generated, until it is delivered to the client.

The design of optimum schedules for data broadcast has attracted a large body of research (see e.g., [4]–[10] and references therein). The prior works in this area typically assume that client requests are distributed uniformly over time and focus on minimizing the *average* waiting time. However, in many settings of practical interest, it is more important to minimize the *worst case* waiting time, i.e., the maximum waiting time experienced by a client, independently of his access pattern. This requirement is typically mandated by the service-level agreements (SLAs) that guarantee a certain bound on the time required to obtain information. Accordingly,

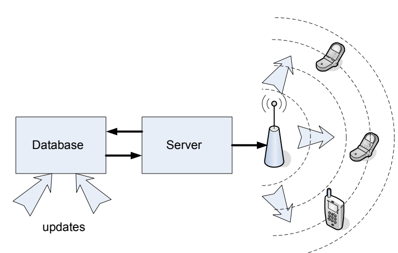

Fig. 1. A typical data broadcast system.

This work was supported in part by the Caltech Lee Center for Advanced Networking and by NSF grants ANI-0322475 and CCF-0346991.

in our previous paper [11], we introduced a notion of *universal* schedules that guarantee low waiting time for any client, regardless of the access pattern. The idea is to model the data distribution process as a game against an adversary. The goal of the adversary is to place "bad" requests, which will result in high values of waiting time. Clearly, a schedule that performs well against such a powerful adversary will perform well for any client's behavior.

In this paper we show that the waiting time can be significantly improved by adding redundancy to the schedule. In particular, each packet is transmitted in an encoded form over a longer time interval than that required for the transmission of a packet without encoding.

A key point in the design of schedules with redundant transmissions is to ensure that the received information is always up-to-date, i.e., has small staleness. In this work we focus on the design of schedules that yield low waiting time for any given staleness constraint. We show a trade-off between the achievable staleness and waiting time in the design of universal schedules. The trade-off has a surprising behavior we refer to as the "knee" phenomenon: For small values of staleness, the minimum waiting time decreases drastically with only a minor increase in staleness; however, after a certain point, any increase in staleness results in only a minor decrease of waiting time (see Fig. 3).

The rest of the paper is organized as follows. In Section II, we formally define our model. In Section III, we describe optimal and approximate schedules that minimize waiting time subject to a given staleness constraint. Finally, in Section IV, we present concluding remarks and directions for future work. Due to space limitations, some proofs and technical details are omitted and can be found in [12].

II. Model

In this paper, we focus on settings in which the broadcast channel is dedicated to a single information source. In such settings, each packet carries the same information, e.g., stock quotes. The content of each packet, however, is different, because each packet captures the most recent state of the information source. We also assume that all packets have an identical size and that the transmission of a packet (without encoding) requires one time unit.

A broadcast schedule specifies the times at which the packets are generated and transmitted. Each packet is allocated a time interval whose length is at least one time unit. Each packet is periodically broadcasted (in correct bit order) over a corresponding time interval. This simple encoding allows the client to restore the original packet from any portion of the interval whose length is at least one time unit.

Definition 1 (Schedule $\mathcal{S}$): A *schedule* is a sequence $\{X_1, X_2, \dots\}$, $X_i \geq 0$, such that $X_i + 1$ specifies the length of the time interval allocated for packet i.

A schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ can also be defined by its *transmission sequence* $\{T_1, T_2, \dots\}$, where T_n represents the beginning of the n^{th} interval, that is, $T_1 = 0$ and $T_n = \sum_{i=1}^{n-1} X_i + n - 1$ for all $n > 1$.

Example 1: Fig. 2(a) depicts schedule $\mathcal{S}_1 = \{0, 0, 0, \dots\}$. In this schedule packet i is transmitted over the interval $[i, i+1]$, for $i = 0, 1, \dots$. Fig. 2(a) depicts a schedule

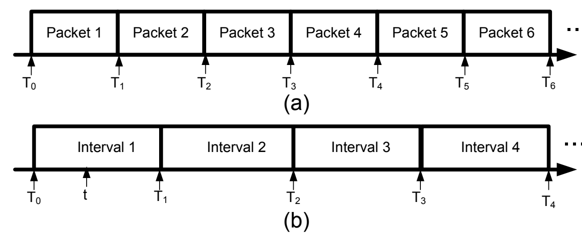

Fig. 2. Examples of possible schedules.

$\mathcal{S}_2 = \{h, h, h, \dots\}$, where $h > 0$. While in the first schedule each packet is sent without encoding, in the second schedule each packet is periodically broadcasted over an interval of length $h + 1$.

A. Waiting Time

Our goal is to design broadcast schedules that minimize waiting time, i.e., the amount of time spent by the client waiting for data. Let $\mathcal{S}$ be a schedule, and suppose that a client request is placed at time t. Also, let n be the *current* interval, i.e., the interval for which $T_n \leq t < T_{n+1}$. The waiting time depends on the time left in the current interval, i.e., $T_{n+1} - t$. Specifically, if $T_{n+1} - t \geq 1$ then the client request can be satisfied within the current interval, hence the waiting time is zero. Otherwise, the client must wait until the beginning of the next interval, hence its waiting time is $T_{n+1} - t$.

Definition 2 (Waiting Time, $WT(\mathcal{S}, t)$): The *Waiting Time* $WT(\mathcal{S}, t)$ for a request at time t using a schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ is defined as follows.

$$WT(\mathcal{S}, t) = \begin{cases} T_{n+1} - t & \textbf{\textit{if}} \quad T_{n+1} - t < 1 \\ 0 & \textit{otherwise,} \end{cases} \tag{1}$$

where n is the current interval, i.e., the interval for which it holds that $T_n \leq t < T_{n+1}$.

For example, in schedule $\mathcal{S}_1$, the waiting time of the request that arrives at time t is $\lceil t \rceil - t$. In contrast, in schedule $\mathcal{S}_2$, the waiting time is zero for many requests. Indeed, suppose that the client arrives at time t, $T_0 \leq t \leq T_1$, as depicted in Fig. 2(b). If the remainder of the current interval is more than one unit, i.e., $T_1 - t \geq 1$, then the waiting time of the client is zero. Otherwise, the client must wait $T_1 - t$ time units for the beginning of the next interval.

B. Staleness

The staleness captures the age of the information delivered to the client. The staleness depends on both the amount of time that has passed from the beginning of the current interval n, i.e., $t - T_n$, and the amount of time left in the current interval, i.e., $T_{n+1} - t$. Specifically, if $T_{n+1} - t \geq 1$ then the client request can be satisfied within the current interval. In this case the client receives the data $t - T_n$ time units after it was obtained from the database, hence the staleness is $t - T_n$. If $T_{n+1} - t \geq 1$, then the client must wait to the beginning of the next interval, and the information it receives will be up-to-date, i.e., the staleness will be zero.

Definition 3 (Staleness, $ST(\mathcal{S}, t)$): The *Staleness* $ST(\mathcal{S}, t)$ for a request at time t using a schedule $\mathcal{S} = \{X_1, X_2, \dots\}$ is defined as follows.

$$ST(\mathcal{S}, t) = \begin{cases} 0 & \textbf{\textit{if}} \quad T_{n+1} - t < 1 \\ t - T_n & \textit{otherwise,} \end{cases} \tag{2}$$

where n is the current interval, i.e., the interval for which it holds that $T_n \leq t < T_{n+1}$.

Note that in the schedule $\mathcal{S}_1$ the staleness is zero for any request. In contrast, in schedule $\mathcal{S}_2$, the staleness is positive for some requests. Indeed, suppose that the client arrives at time t, $T_0 \leq t \leq T_1$ (see Fig. 2(b)). If the remainder of the current interval is more than one unit, i.e., $T_1 - t \geq 1$, then the client request is satisfied within the current interval and the staleness is $t - T_0$. Otherwise, the client must wait for the beginning of the next interval, hence the staleness is zero.

We note that staleness and waiting time have certain duality properties. Namely, for any specific value of t it is the case that exactly one of the two is non-zero. In general, as we show below, a lower waiting time can be achieved at a cost of higher staleness. In this paper we study this trade-off in the context of universal schedules.

C. Universal Schedules

The goal of universal schedules is to minimize waiting time for any client, regardless of its behavior. For this purpose, we assume that the requests are generated by an adversary, whose purpose is to generate requests resulting in high waiting time. We note that both schedules S_1 and S_2 have a poor performance in the presence of an adversary. Indeed, suppose that an adversary puts its request at time $t = T_i - 1 + \Delta$, for some small value of $\Delta > 0$, where T_i is the beginning of some interval i. In this case, the waiting time is $1 - \Delta$, which can be arbitrary close to one time unit.

We observe that the worst-case waiting time of any deterministic schedule is close to one time unit[1]. Indeed, since the adversary knows the schedule, it can generate a request $1 - \Delta$ units of time before the transmission of the next item. A natural way to deal with such a powerful adversary is by adding randomness to the schedule. In a random schedule, the lengths X_i of all intervals are random variables. This implies, in turn, that the values of waiting time $WT(\mathcal{S}, t)$ and staleness $ST(\mathcal{S}, t)$ for any request time t are also random variables.

D. Expected staleness and waiting time

In randomized settings, there are several types of adversaries that can be considered [13]. One type is an *oblivious* adversary, i.e., an adversary that decides about its requests in advance, before the broadcast begins. This adversary is relatively week and can be dealt with by transmitting an empty interval of random length followed by a deterministic schedule [11]. In this paper we assume that the adversary is *adaptive*, i.e., a request generated at time t is based on the *history* of the schedule from the beginning of the transmission up to time t. Such an adversary models the worst possible access pattern, including possible correlations between requests and past transmissions.

In order to define the expected staleness and waiting time for adaptive adversaries we condition the probability distribution of a given random schedule $\mathcal{S}$ on the history of $\mathcal{S}$ up to time t. Intuitively, the history of a schedule can be described by the lengths of the intervals transmitted up to time t.

[1]The only exception is schedule in which one packet is broadcasted over an infinite interval, i.e., schedule $\mathcal{S}_2$ with $h = \infty$. This schedule, however, has unbounded staleness and hence cannot be used for practical purposes.

Definition 1: A history $H = (t, x_1, x_2, \ldots, x_l)$ of a random schedule $\mathcal{S} = \{X_1, X_2, \ldots\}$ at time t is the event in which (a) For all $i, 1 \leq i \leq l$, it holds that $X_i = x_i$; and (b) $\sum_{i=1}^{l} X_i + l \leq t < \sum_{i=1}^{l+1} X_i + l + 1$.

In other words, $H = (t, x_1, x_2, \ldots, x_l)$ is the event in which (a) For the first l random variables in $\mathcal{S}$ it holds that $X_i = x_i$, and (b) The number of intervals that are completely broadcasted up to time t is l.

Formally, let H be a history event. H is said to be admissible if the probability that it occurs is non-zero. For admissible histories H, let $\mathcal{S}|H$ be the schedule obtained by conditioning $\mathcal{S}$ on the event H. Notice that $\mathcal{S}|H$ is also a random schedule. For any request time t, our objective is to obtain bounds on the expected waiting time and staleness of a schedule regardless of the history of the schedule up to time t. Accordingly, the worst-case expected waiting time of a schedule is defined as follows:

$$EWT(\mathcal{S}) = \sup_{H,t} E[WT(\mathcal{S}|H, t)]. \tag{3}$$

Similarly, the worst-case expected staleness is defined by:

$$EST(\mathcal{S}) = \sup_{H,t} E[ST(\mathcal{S}|H, t)]. \tag{4}$$

Here, the expectation is over the schedule distribution $\mathcal{S}|H$, and we are maximizing over admissible history events H.

Example 2: Consider the schedule in which the length of each interval is uniformly distributed on $[1, 2]$. It can be easily verified that the worst-case expected waiting time of this schedule is 0.5, which is a significant (50%) improvement over deterministic schedules. A simple calculation shows that the worst-case expected staleness of this schedule is just 0.25. As we show below, a lower waiting time can be achieved under the same staleness constraint.

E. I.i.d. schedules

In the rest of this study we focus on schedules $\mathcal{S} = \{X_1, X_2, \ldots\}$ in which all random variables X_i are independent and identically distributed (i.i.d.), that is $X_i = X$ for all i. Such schedules are referred to as i.i.d. schedules. In the full version of this paper [12], we show that for any schedule $\mathcal{S}$ there exists an i.i.d. schedule $\mathcal{S}'$ which is at least as good as $\mathcal{S}$, i.e., $EWT(\mathcal{S}') \leq EWT(\mathcal{S})$ and $EST(\mathcal{S}') \leq EST(\mathcal{S})$. This implies that our i.i.d. assumption does not result in any loss of generality.

F. Problem formulation

In this study we investigate the following problem: Given a staleness constraint s, find a schedule $\mathcal{S}$ whose worst-case expected waiting time $EWT(\mathcal{S})$ is minimal subject to the staleness constraint $EST(\mathcal{S}) \leq s$.

We denote by $OPT(s)$ the minimum worst-case expected waiting time of a schedule that satisfies staleness constraint s.

III. Optimum Schedules

A. Maximum waiting time and staleness

Let X be a random variable, and let F be its cumulative distribution function. In this section we represent the expected waiting time and staleness yielded by an i.i.d. schedule $\mathcal{S}$ defined by X in terms of F.

Theorem 1: Let $\mathcal{S}$ be an i.i.d. schedule with distribution $F(x)$. Also let t be a request time and let $H = (t, x_1, x_2, \ldots, x_l)$ be the history of $\mathcal{S}$ at time t. We denote by n the current interval, i.e., the interval for which $T_n \le t < T_{n+1}$ and by τ the time that has passed from the beginning of current interval, i.e., $\tau = t - T_n$. Then, the expected waiting time $E[WT(\mathcal{S}|H,t)]$ for this request is

$$E[WT(\mathcal{S}|H,t)] = \begin{cases} F^-(\tau) - \int_0^\tau F(x)dx & \textbf{\textit{if}} \quad \tau < 1 \\ \frac{F^-(\tau) - \int_{\tau-1}^{\tau} F(x)dx}{1-F(\tau-1)} & otherwise, \end{cases} \tag{5}$$

where $F^-(x) = P(X < x)$.

Proof: Let $\hat{X}$ be the random variable that describes the length of the current interval conditioned on the viewed history H. We also denote by $\hat{F}(x)$ the cumulative distribution function of $\hat{X}$. Since the length of the current interval is at least τ, it holds that $\hat{X} = X_n \mid \{X_n + 1 > \tau\}$. Note that for $\tau \le 1$, the adversary has no knowledge about X_n, hence, in this case $\hat{X} = X_n$ and $\hat{F}(x) = F(x)$. For $\tau > 1$ it holds that $\hat{F}(x) = \frac{F(x)-F(\tau-1)}{1-F(\tau-1)}$.

The waiting time $WT(\mathcal{S}|H,t)$ depends on the time remained in the current interval, i.e., $\hat{X} + 1 - \tau$. We consider the two following cases:

1) **Case 1:** $\hat{X} - \tau + 1 \ge 1$ or $\hat{X} \ge \tau$. In this case the remaining time in the interval is one unit of time or more. Thus, according to Equation (1), the waiting time is zero, i.e., $WT(\mathcal{S}|H,t) = 0$.
2) **Case 2:** $\hat{X} + 1 - \tau < 1$ or $\hat{X} < \tau$. In this case, the client has to wait until the beginning of the next interval. According to Equation (1), the waiting time in this case is $WT(\mathcal{S}|H,t) = \hat{X} + 1 - \tau$.

We proceed to derive the expression for the expected waiting time $E[WT(\mathcal{S}|H,t)]$. We begin with the case $\tau < 1$ and identify the distribution function $F_{WT}(x)$ of $WT(\mathcal{S}|H,t)$. We consider the following cases:

1) $x = 0$. We note that the waiting time is zero if and only if $\hat{X} \ge \tau$. Thus, $F_{WT}(0) = P(\hat{X} \ge \tau) = 1 - \hat{F}^-(\tau) = 1 - F^-(\tau)$.
2) $0 < x < 1-\tau$. We note that for $\tau < 1$ the waiting time is either zero or at least $1-\tau$. Hence for $0 < x < 1-\tau$ it holds that $F_{WT}(x) = F_{WT}(0) = 1 - F^-(\tau)$.
3) $1 - \tau \le x < 1$. Then, $F_{WT}(x) = P(WT(\mathcal{S}|H,t) \le x) = P(WT(\mathcal{S}|H,t) < 1 - \tau) + P(1 - \tau \le WT(\mathcal{S}|H,t) \le x) = F_{WT}(0) + P(1-\tau \le \hat{X}+1-\tau \le x) = F_{WT}(0) + P(0 \le \hat{X} \le x+\tau-1) = 1 - F^-(\tau) + F(x+\tau-1)$.

Notice that for $x \ge 1$, $F_{WT}(x) = 1$. We are ready to compute the expected waiting time $E[WT(\mathcal{S}|H,t)]$.

$$E[WT(\mathcal{S}|H,t)] = \int_0^\infty (1 - F_{WT}(x))dx =$$
$$= F^-(\tau) - \int_{1-\tau}^{1} F(x+\tau-1)dx = F^-(\tau) - \int_0^\tau F(x)dx$$

Now we consider the case in which $\tau \ge 1$. There are two possible cases.

1) $x = 0$. In this case $F_{WT}(0) = P(\hat{X} \ge \tau) = 1 - \hat{F}^-(\tau) = 1 - \frac{F^-(\tau)-F(\tau-1)}{1-F(\tau-1)}$.
2) $0 < x < 1$. In this case $F_{WT}(x) = P(WT(\mathcal{S}|H,t) \le x) = P(0 < WT(\mathcal{S}|H,t) \le x) + P(WT(\mathcal{S}|H,t) = 0) = P(0 < \hat{X} + 1 - \tau \le x) + F_{WT}(0) = P(\tau - 1 < \hat{X} \le x+\tau-1) = \hat{F}(\tau+x-1) - \hat{F}(\tau-1) + F_{WT}(0) = \frac{F(\tau+x-1)-F(\tau-1)}{1-F(\tau-1)} + F_{WT}(0)$.

We now compute the expected waiting time $E[WT(\mathcal{S}|H,t)]$:

$$E[WT(\mathcal{S}|H,t)] = \int_0^\infty (1 - \hat{F}(x))dx =$$
$$= \frac{F^-(\tau) - \int_0^1 F(\tau+x-1)dx}{1-F(\tau-1)} = \frac{F^-(\tau) - \int_{\tau-1}^{\tau} F(x)dx}{1-F(\tau-1)}.$$

■

Theorem 2: Let $\mathcal{S}$ be an i.i.d. schedule with distribution $F(x)$. Let t be a request time and let $H = (t, x_1, x_2, \ldots, x_l)$ be the history of $\mathcal{S}$ at time t. We denote by n the current interval, i.e., the interval for which $T_n \le t < T_{n+1}$ and by τ the time that has passed from the beginning of current interval, i.e., $\tau = t - T_n$. Then, the expected staleness $E[ST(\mathcal{S}|H,t)]$ for this request is

$$E[ST(\mathcal{S}|H,t)] = \begin{cases} \tau(1 - F^-(\tau)) & \textbf{\textit{if}} \quad \tau < 1 \\ \tau\left(\frac{1-F^-(\tau)}{1-F(\tau-1)}\right) & otherwise, \end{cases} \tag{6}$$

where $F^-(x) = P(X < x)$.

The proof follows the same lines as the proof of Theorem 1. For more details, see [12].

Theorems 1 and 2 imply that the expected staleness and waiting time depend only on the time τ that has passed since the beginning of the current interval, i.e., the history of the schedule $H = (t, x_1, x_2, \ldots, x_l)$ up to time t can be summarized by a single parameter τ. This allows to compute the worst-case expected waiting time and staleness by finding values of τ that maximize Equations 5, 6, respectively.

B. Optimal solution for small values of staleness

In this section we focus on the special case in which the values of the staleness constraint s are *small*, i.e., $s \le 0.13$. For such values of s, we obtained a closed form *optimal* solution for the problem at hand.

Theorem 3: Let $s \le 0.13$ be a staleness constraint. Then, the optimal schedule that satisfies s has distribution function $F(x) = \min\{1, C(s)e^x\}$ and yields worst-case expected waiting time $C(s)$, where $C(s) = \left(1 - \frac{2s}{s+\sqrt{s(4+s)}}\right) e^{-\frac{1}{2}(s+\sqrt{s(4+s)})}$.

The proof of Theorem 3 is omitted and can be found in [12].

C. Approximation algorithm

In this section we turn to find (almost) optimal schedules for a given value of staleness constraint s. We present an approximation algorithm, which receives as input a staleness constraint s and any (arbitrarily small) approximation parameter ε, and returns a schedule $\mathcal{S}$ whose worst-case expected staleness is at most s and whose worst-case expected waiting time is at most $OPT(s) + \varepsilon$. The computational complexity of our algorithm is polynomial in $\frac{s}{\varepsilon}$.

Our approximation has two steps. First, we show that for any $\varepsilon_1 > 0$ there exists a schedule $\mathcal{S}_1$ such that $EST(\mathcal{S}_1) \le s$,

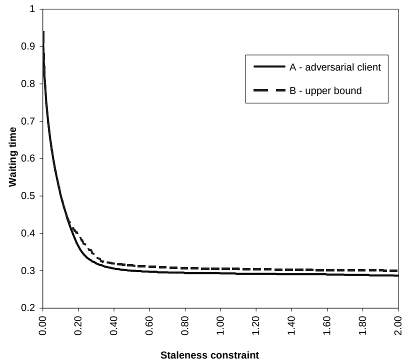

Fig. 3. Trade-off between staleness and waiting time for adversarial clients (A) and an upper bound on this trade-off using the explicit distributions described in Section IV (B).

$EWT(\mathcal{S}_1) \leq OPT(s)+\varepsilon_1$ and the support[2] of $\mathcal{S}_1$ is bounded by $\frac{s}{\varepsilon_1}$. In other words, the optimal distribution can be approximated by a distribution with bounded support. Second, we show that for any $\varepsilon_2 > 0$ the schedule $\mathcal{S}_1$ can be approximated by a schedule $\mathcal{S}_2$ whose distribution is a piecewise-constant function that includes at most $\frac{s}{\varepsilon_1\varepsilon_2}$ segments. This schedule satisfies the staleness constraint, i.e., $EST(\mathcal{S}_2) \leq s$, and its maximum waiting time is more than that of $\mathcal{S}_1$ by at most ε_2, i.e., $EWT(\mathcal{S}_2) \leq EWT(\mathcal{S}_1) + \varepsilon_1 \leq OPT(s) + \varepsilon_1 + \varepsilon_2$. Moreover, we construct a Linear Program that computes $\mathcal{S}_2$. The running time of this program is polynomial in $\frac{s}{\varepsilon_2}$. As a result, for any $\varepsilon > 0$ we can compute a schedule that satisfies the staleness constraint s and whose maximal waiting time is at most $OPT(s) + \varepsilon$. Indeed, by setting $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2}$, we ensure that the schedule $\mathcal{S}_2$ satisfies the above requirements.

Theorem 4: There exists an efficient algorithm, that for a given staleness constraint s and approximation parameter $\varepsilon > 0$, computes a schedule $\mathcal{S}$ that satisfies $EST(\mathcal{S}) \leq s$ and $EWT(\mathcal{S}) \leq OPT(s) + \varepsilon$. The computational complexity of the algorithm is polynomial in $\frac{s}{\varepsilon}$.

Due to space limitations, the LP formulation and the proof of Theorem 4 appear in [12].

D. Numerical results

We used the optimal and approximation algorithm presented in the previous sections in order to compute the attainable values of worst-case waiting time for a broad range of staleness constraints. Our results establish a trade-off between the staleness and waiting time of universal broadcast schedules. The trade-off is depicted on Fig. 3 (series A). This trade-off has a surprising behavior we refer to as the “knee” phenomenon: for small values of staleness (typically below 0.3) the minimum waiting time decreases drastically with only a minor increase in the staleness constraint; however, for large values of the staleness constraint (above 0.3), any increase in the staleness constraint results in only a minor decrease of waiting time. A direct result of the knee phenomenon is the existence of a schedule that has small maximum expected waiting time (0.31) and whose worst-case expected staleness is also small (at most 0.3). This point represents a reasonable trade-off between waiting time and staleness. The corresponding schedule reduces the worst-case waiting time by 70% compared to a deterministic schedule while ensuring that the distributed information is up-to-date.

[2]The support of a Cumulative Distribution Function $F(x)$ is a set of values of x at which the function is strictly less than 1, i.e., $\{x \mid F(x) < 1\}$.

IV. CONCLUSION

In this paper, we have studied the design of optimal universal schedules for discrete broadcast. We have defined the notion of staleness, and have presented a tight characterization of the minimal waiting time for a given staleness constraint. Our results are optimal for small values of staleness, and arbitrarily close to the optimum for general staleness values.

The study of analytical (closed form) approximate solutions to the problem at hand gave rise to the following empirical observation. For arbitrary values of s, the distribution function $G_s(x) = 1-\Gamma(s+a(s))\frac{s^{x+1-s-a(s)}}{\Gamma(x+1)}$ yields worst-case waiting time which is very close to optimum. Here $a(s)$ is a constant between 0 and 1, and Γ represents the standard Gamma function. The staleness/waiting time trade-off of our schedules defined by $G_s(x)$ are depicted in Figure 3 (series B).

Many questions in the setting of universal broadcasting remain open. First, in our framework we have assumed that the packets are transmitted over a channel without errors. The next step would be to consider lossy communication channels. In such settings, we need to change the repetition encoding of packets to one that admits error correction. Another research direction is to study optimal schedules for multiple information sources. We believe that the tools presented in this work lay the foundation for dealing with a wide range of related problems.

REFERENCES

[1] T. Imielinski, S. Viswanathan, and B. Badrinath, “Data on Air: Organization and Access,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, no. 3, pp. 353–372, 1997.

[2] D. Cheriton, “Dissemination-Oriented Communication Systems,” Stanford University, Tech. Rep., 1992.

[3] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Broadcast Disks: Data Management for Asymmetric Communication Environments,” in *Proceedings of ACM SIGMOD*, San Jose, CA, May 1995, pp. 199–210.

[4] M. Ammar, “Response Time in a Teletext System: An Individual User’s Perspective,” *IEEE Trans. Comm.*, vol. COM-35, no. 11, pp. 1159–1170, 1987.

[5] M. H. Ammar and J. W. Wong, “The Design of Teletext Broadcast Cycles,” *Performance Evaluation*, vol. 5, no. 4, pp. 235–242, 1985.

[6] S. Hameed and N. H. Vaidya, “Log-time Algorithms for Scheduling Single and Multiple Channel Data Broadcast,” in *Proceedings of MOBICOM’97*, 1997.

[7] N. H. Vaidya and S. Hameed, “Data Broadcast in Asymmetric Wireless Environments,” in *Proceedings of First International Workshop on Satellite-based Information Services (WOSBIS)*, 1996.

[8] K. Foltz, L. Xu, and J. Bruck, “Scheduling for Efficient Data Broadcast over Two Channels,” in *Proceedings of International Symposium on Information Theory (ISIT)*, Chicago, IL, USA, June 2004, p. 113.

[9] A. Bar-Noy, J. Naor, and B. Schieber, “Pushing Dependent Data in Clients-Providers-Servers Systems,” *Wireless Networks*, vol. 9, no. 5, pp. 421–430, 2003.

[10] Z. Brakerski, A. Nisgav, and B. Patt-Shamir, “General Perfectly Periodic Scheduling,” in *Proceedings of PODC ’02*, June 2002, pp. 163–172.

[11] M. Langberg, A. Sprintson, and J. Bruck, “Optimal Universal Schedules for Discrete Broadcast,” in *Proceedings of International Symposium on Information Theory (ISIT)*, Chicago, IL, USA, June-July 2004.

[12] ——, “Optimal Schedules for Asynchronous Transmission of Discrete Packets,” California Institute of Technology, October,” ETR062, 2004, available from: http://www.paradise.caltech.edu/ETR.html.

[13] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.